Future Generation Computer Systems 92 (2019) 777–788

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A social-based watchdog system to detect selfish nodes in opportunistic mobile networks

Behrouz Jedari [a], Feng Xia [a,*], Honglong Chen [b], Sajal K. Das [c], Ami
Zafer AL-Makhadmeh [d]

[a] School of Software, Dalian University of Technology, Dalian 116620, China
[b] College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China
[c] Computer Science Department, Missouri University of Science and Technology, USA
[d] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[e] Mathematics Department, Faculty of Science, Menoufia University, Shebin El-Kom 32511, Egypt

引用他的论文 /很新/相关性很高/很有意义。。
IS 是单独自私性node：自己的为1，否则根据message从此中继到真正dest的延迟决定
SS是社会自私性node：考虑到本SS节点和src、sender的社会关系强度，
按照IS/SS节点、social强度来计算utility值，先转发utility高的message,

计算 αi，wch节点更新对FC全合作/IS单独自私/SS社会自私的概率评价【直接评价direct evaluation】【间接评价indirect evaluation】
筛选 间接概率评价 偏差过大
\sum|三种概率评价的偏差| <1

df 正面证据 和 负面证据 传播因子（随机决定要不要传输）
检测效果
benchmark [19]
IEEE Transactions on Information Forensics and Security (Volume: 7 , Issue: 2 , April 2012)
Mitigating Routing Misbehavior in Disruption Tolerant Networks

altrusim 利他主义

HIGHLIGHTS

- A novel altruism model to realize the individual and social selfish behavior of nodes is devise
- A watchdog mechanism to analyze node behavior with respect to routing utility is proposed
- A reputation system to detect selfish nodes and identify their selfishness degree is designed.
- Extensive simulations illustrate the effectiveness and efficiency of the proposed scheme.

ARTICLE INFO

Article history:
Received 16 June 2017
Received in revised form 8 October 2017
Accepted 29 October 2017
Available online 6 November 2017

Keywords:
Opportunistic mobile networks
Cooperative routing
Selfish behavior
Game theory
Incentive scheme

ABSTRACT

Detecting selfish nodes in opportunistic mobi
thus improve the data delivery performance. M
nodes' contact records and do not consider the
behavior, which result in long detection time an
distinguish the nodes' selfishness type and degr
mechanisms applied to stimulate different no
Social-based Watchdog system (SoWatch) in v
their encountered nodes with respect to their so
in message relaying. Meanwhile, the watchdog nodes apply the second-hand watchdog information received from other nodes to improve the detection time and accuracy. Next, we design a reputation system in which watchdog nodes identify selfish nodes based on their direct and indirect watchdog information and distinguish individually and socially selfish nodes. Furthermore, we design a watchdog evaluation module to protect SoWatch against wrong watchdogs disseminated by malicious nodes in which a watchdog node investigates the truthfulness of the indirect watchdogs before applying them. Our experiments using real-world datasets illustrate that SoWatch outperforms a benchmark contact-based watchdog system in terms of detection time by 45% and detection ratio by 10% with less communication overhead.

1. Introduction

Opportunistic Mobile Networks (OMNs) are novel communication paradigm in which mobile nodes (*i.e.*, users and their devices) can directly exchange data among each other via short-range wireless technologies (*e.g.*, Bluetooth or Wi-Fi). In general, OMNs are employed in scenarios when the nodes do not have access to the Internet due to some reasons such as the limited coverage of cellular networks or quality-of-service requirements of data-intensive mobile applications. In OMNs, Delay-Tolerant Networks (DTNs) [1] are considered as the core architecture, based on which the nodes employ *store–carry-and-forward* fashion to exchange data among each other. OMNs have many applications in different areas such as mobile data offloading [2], vehicular social networks [3], and emergency scenarios [4].

* Corresponding author.
E-mail address: f.xia@ieee.org (F. Xia).

https://doi.org/10.1016/j.future.2017.10.049
0167-739X/

The majority of existing routing protocols in OMNs suppose that nodes are *Fully Cooperative (FC)* and they follow the principles of the underlying routing protocol [5,6]. Thus, they explore the nodes' contact history, social, and context information to improve the network throughput. However, some nodes may exhibit selfish behavior and refuse to carry messages on behalf of others due to various reasons such as resource limitations or social preferences. In general, two types of selfish nodes can exist in OMNs [7–9]: *Individually Selfish (IS)* and *Socially Selfish (SS)*. IS nodes mainly forward their own messages but refuse to relay messages for other nodes, whereas SS nodes mitigate the degree of their selfishness based on their social ties and relay messages received from those with whom they have strong social ties.

The impact of IS and SS nodes on the performance of opportunistic routing protocols has been studied in [10–12] and [13,14], respectively. Overall, it is demonstrated that the routing performance is degraded significantly if a huge number of nodes exhibit selfish behavior. For instance, the authors in [12] show that when 30% of highly-connected nodes drop messages, SimBet protocol [15] delivers only 3% of messages and wastes 95% of the transmissions. The impact of the nodes' non-cooperative behavior on routing can be more harmful when malicious nodes drop their received messages but produce forged metrics (blackhole and greyhole attacks [16]) to disrupt the routing process or attract more messages. Thus, it is indispensable to detect selfish nodes and act against malicious nodes.

一致性检查

Several distributed detection schemes have been proposed for OMNs in which the nodes investigate the consistency of contact records collected from different nodes to detect selfish message droppers [16–18]. Some existing works (*e.g.,* [19–22]) employ watchdog mechanisms to efficiently detect the selfish nodes in which some trusted watchdog nodes analyze the properties of messages received from their encounters to decide whether a node is selfish (positive detection) or not (negative detection). However, the watchdog nodes may not have sufficient direct watchdog information about other nodes because the inter-contact time (*i.e.,* two consecutive contacts) between the nodes in OMNs can be quite long. Hence, they can exchange their watchdog information with each other in order to detect the selfish nodes swiftly and accurately. The performance of a watchdog system can be severely degraded if malicious nodes generate wrong watchdog information about other nodes so that the watchdog nodes may detect cooperative nodes as selfish (false positive) or selfish nodes as cooperative (false negative). Thus, an effective mechanism should be designed to protect the watchdog system against the false positives and negatives. Once selfish nodes are detected, an incentive scheme (*e.g.,* [23,24]) or an exclusion method (*e.g.,* [16]) can be applied to mitigate their selfish behavior.

Almost all the above-mentioned selfish node detection schemes apply the nodes' contact and message records to detect selfish nodes that result in long detection time and high communication overhead. Meanwhile, they cannot distinguish IS and SS nodes. Although a recent proposal (SENSE [25]) has used the nodes' social information to detect selfish nodes, it only considers binary social relationships between the nodes in which two nodes are either socially connected or not. However, selfish users usually alleviate their selfishness level based on the strength of their social ties [7]. Meanwhile, nodes with strong social ties and similarities contact each other more frequently, and thus they have a high degree of cooperation with each other [26]. Consequently, the nodes' social tie information can help detect selfish nodes efficiently. In addition, the social tie information of nodes can help identify the type of selfish nodes (*i.e.,* IS and SS) and their selfishness degree, which is very useful because the charge and rewarding mechanisms applied to stimulate IS and SS nodes may not be the same [27].

In this paper, we propose a social-based watchdog system, namely SoWatch, in which watchdog nodes analyze messages received from their encountered nodes based on the social tie information to detect selfish nodes and identify their selfishness type. In SoWatch, a watchdog node calculates the individual and social routing utility of messages sent by its encountered sender node that help the watchdog node to identify the possible selfish behavior of the sender node and its selfish degree. In addition, the watchdog node utilizes indirect watchdog information diffused by its encountered nodes to update its opinion about the selfish behavior of other nodes that help improve the detection time and accuracy. Next, we design a reputation system in which the watchdog node updates its opinion about the other nodes based on the direct and indirect watchdog information and identify their type.

Our major contributions can be summarized as:

- We advocate a novel general altruism model in which selfish nodes mitigate their cooperation in data relaying based on their individual and social preferences.
- We propose a watchdog system in which the watchdog nodes analyze messages based on their individual and social routing utility to the sender nodes to distinguish IS and SS nodes.
- We design an indirect watchdog evaluation module to protect the watchdog system against wrong watchdog information disseminated by malicious nodes in which the watchdog nodes investigate the truthfulness of the indirect watchdogs before applying them.
- We incorporate a reputation system into the watchdog system in which the watchdog nodes employ an effective mechanism to update the reputation of nodes based on their direct and indirect watchdog information.
- Our experiments using real-world datasets illustrate that our proposed watchdog system outperforms a benchmark contact-based watchdog scheme in detection time by 45% and detection ratio by 10% with less communication cost.

We structure the rest of the paper as follows. In Section 2, we review the related work. In Section 3, the system model is presented. In Section 4, we present our proposed watchdog system. In Section 5, we report the evaluation results. We conclude the paper in Section 6.

2. Related work

Several mechanisms have been proposed to detect message droppers in mobile ad hoc networks that can be categorized into two major approaches: neighborhood monitoring and acknowledgment (ACK). Marti et al. [12] is a well-known monitoring system in which the sender of a message verifies whether its encountered node forwards its message or not. However, the nodes in monitoring methods cannot observe the behavior of their second-hop nodes. Thus, Liu et al. [28] proposed a *2ACK* scheme in which the receiver of a message sends *2ACK* packet back to the sender to indicate that the message has been received successfully. However, the techniques discussed above rely on end-to-end connections among nodes, which cannot be applied to OMNs.

Recently, some distributed detection schemes have been designed in OMNs. Li and Cao [19] proposed a detection mechanism in which each node is required to transfer the list of its sent and received messages to its encountered nodes. Thus, each node can check the consistency of the received records from multiple nodes and determine message droppers. Similarly, the nodes in [17] exchange a securely signed encounter ticket every time they contact each other that prevent attackers from claiming non-existing encounters. In addition, Guo et al. [16] employed a contact-based approach in which an adaptive detection threshold is chosen to

maximize detection rate while minimizing false positive detections. However, [17] and [16] cannot detect colluding message droppers. Diep et al. [18] proposed a statistical-based defense scheme, namely SDBG, to detect both individual and colluding nodes in which the nodes exchange their contact history with each other that let a node to judge the behavior of other nodes. Once an individual attacker is detected, SDBG starts detecting possible colluding attackers based on the number of messages received from the individual attacker.

Unlike the works above, the authors in [29] proposed a collaborative watchdog scheme in which watchdog nodes use the direct and indirect watchdog information to discover selfish nodes. The system includes three modules: Watchdog module that detects selfish nodes and new contacts; Diffusion module that diffuses positive and negative detections; and Network Information module that includes a transition machine to update the status of nodes in which a watchdog node has one of the three statuses *Positive*, *Negative*, and *NoInfo* about other nodes. In addition, a Markov chain model is designed to calculate the detection time and ratio. The extension of [29] is *CoCoWa* [21] in which a reputation scheme is designed to protect the detection against malicious behavior. The evaluation results demonstrate that *CoCoWa* reduces the detection time, ranging from 20% for a low degree of collaboration to 99% for higher levels of collaboration.

Zhu et al. [20] proposed *iTrust* in which a Trusted Authority (TA) analyzes the nodes' forwarding behavior based on their contact evidence periodically. Particularly, a game-theoretic approach was applied to demonstrate that TA could achieve the tradeoff between the accuracy and detection cost. Furthermore, a reputation system is designed, based on which a node with a good reputation is checked with a lower probability while a bad reputation node is checked more frequently. However, *iTrust* depends on a centralized TA that may not be accessible in realistic OMNs. Similarly, cooperative watchdog system [22] assigns a reputation to each node that is updated based on their cooperation level. When two nodes contact each other, they exchange their opinions about the reputation of their previous encountered nodes. Based on the reputation score, the status of a neighbor node can be cooperative, partial cooperative, neutral, suspected, and selfish. While the methods mentioned above can detect whether a node is either cooperative or selfish, our proposed scheme can distinguish the type of selfish nodes (*i.e.,* IS and SS) and identify their selfishness degree.

3. System model

3.1. Network model

Similar to [30], we consider the network in two domains: a physical domain and a social domain. In the following, we define the essential features of each domain.

(1) *Physical Domain:* we model the physical domain as a general DTN with N mobile nodes in which two nodes contact each other when they come within the communication range of each other. Meanwhile, they sporadically have access to online social network servers via Wi-Fi to update their social features. When nodes $i, j \in N (i \neq j)$ encounter each other at time t, each one generates a *Contact Record (CR)* with six main fields: $CR^t_{i,j} = \langle i, j, CID_i, CID_j, S_i, R_i \rangle$ where i and j are the identity of nodes i and j, respectively. Furthermore, CID_i and CID_j are the unique sequence numbers assigned to $CR^t_{i,j}$ by nodes i and j for integrity protection. We suppose that CID_i of i's new CR equals the sequence number of its latest CR incremented by 1. In addition, S_i and R_i are the lists of messages i forwarded to and received from j during the contact, respectively.

To provide secure communications, we suppose that there exists an anonymous identity-based authentication service in which each node has a pair of public and private keys. In this way, the nodes only know the public key of each other and sign their CRs using their public key. Thus, the nodes cannot forge their identity to launch Sybil attack [31].

(2) *Social Domain:* we use a time-varying graph structure $G=\{G^0, G^1, \ldots, G^t, \ldots\}$ to model the social domain where $G^t = (N, ST)$ represents the graph at time t, N denotes the nodes and $ST = \{(i, j) | i, j \in N, i \neq j\}$ denotes the social ties between them. Each node $i \in N$ stores its social features in feature space $F_i = \{f_{i,1}, f_{i,2}, \ldots, f_{i,k}\}$ where $f_{i,l} (1 \leq l \leq k)$ represents lth feature of i. Furthermore, node i maintains a feature table FT_i to store the feature spaces of other nodes. The strength of social tie between nodes i and j is denoted as $ST_{i,j} \in [0, 1]$ where $ST_{i,j} = 0$ implies that there exists no tie between i and j, while $ST_{i,j} = 1$ stands for the strongest social tie between them. We will explain how to calculate the social tie strength in Section 4.2.

3.2. Node buffer and message model

We assume that each node $i \in N$ has a limited buffer space to store messages, denoted as M_i. Two types of messages can be stored in M_i: *local messages* that are originally generated by i, and *non-local messages* that are received by i for relaying. Since M_i is limited, i may drop a received message if its buffer is full, but it is not considered as i's misbehavior. Each message $m \in M_i$ includes a header with four fields $\langle m_{ID}, m_{src}, m_{des}, m_{ttl} \rangle$ where m_{ID} is the unique identity of m, m_{src} is the source of m, m_{des} is the destination of m, and m_{ttl} is Time-To-Live (TTL) of m that indicates the remaining time before m expires.

3.3. Watchdog node model

We assume that there exist three types of nodes in the network: Watchdog (W), Selfish (S), and Malicious (M) nodes ($|N| = |W| + |S| + |M|$) where the type of each node is stable over time. W nodes normally participate in data relaying by following the underlying routing protocol. Meanwhile, they apply the direct and indirect watchdogs to detect S nodes and identify their selfishness type (*i.e.,* IS and SS). We assume each W node wch has a watchdog table to store its opinion about the reputation of other nodes where $P^{wch}_{i,FC}$, $P^{wch}_{i,IS}$, $P^{wch}_{i,SS}$ denote the probability that i is Fully Cooperative (FC), Individually Selfish (IS), or Socially Selfish (SS) from wch's point of view.

3.4. Selfish node model

Mobile carriers in realistic OMN scenarios mitigate the degree of their selfishness in data relaying based on their individual and social benefits [32]. Based on this idea, we design a flexible altruism model in which two types of S nodes can be defined as follows:

(1) *Individually Selfish (IS) node:* an IS node forwards its local messages. Meanwhile, it stores and relays non-local messages that have a high probability to be delivered to its destination by this node. To realize this behavior, we define $U^{Ind}_{sel}(m) \in [0, 1]$ as the individual utility of message m to IS node $sel \in S$. When sel contacts another node, it replicates its messages in M_{sel} to another node in a descending order of $U^{Ind}_{sel}(m)$, $m \in M_{sel}$, which implies that a message with the highest individual utility is forwarded first. As a receiver, sel accepts storing and relaying m if $U^{Ind}_{sel}(m) > \partial$ where ∂ is a threshold value that is identified by the device carrier. Otherwise, sel drops m but generates a fake CR by reusing the sequence number of its previous CRs to hide its message dropping. We calculate the individual utility of relaying m to sel as:

$$U^{Ind}_{sel}(m) = \begin{cases} 1 & \text{if } m_{src} = sel \\ \max\left(0, 1 - \dfrac{E\left[d_{sel,m_{des}}\right]}{m_{ttl}}\right) & \text{otherwise} \end{cases} \quad (1)$$

Table 1
Notations and variables.

Notations	Descriptions
W	the set of watchdog nodes
wch	a sample watchdog node
S	the set of selfish nodes
sel	a sample selfish node
M	the set of malicious nodes
mal	a sample malicious node
$CR^t_{i,j}$	contact record between nodes i and j
F_i, FT_i	the feature space and table of node i
$ST_{i,j}$	the social tie between nodes i and j
M_i	messages in buffer node i
m_{src}, m_{des}	the source and destination of message m
$P^{wch}_{i,FC}$	the FC reputation of i observed by wch
$P^{wch}_{i,IS}$	the IS reputation of i observed by wch
$P^{wch}_{i,SS}$	the SS reputation of i observed by wch
α_{sel}	the social-awareness degree of node sel
$U^{Ind}_i(m)$	the individual utility of message m to i
$U^{Soc}_i(m)$	the social utility of message m to i

where m_{src} is the source of m, $E\left[d_{sel,m_{des}}\right]$ is the expected delivery delay of m to its destination, and m_{ttl} is TTL of m. In Eq. (1), forwarding local message m ($m_{src} = sel$) brings the highest individual utility to sel, whereas relaying a non-local message m brings the highest individual utility to sel if $E\left[d^m_{sel,des}\right]$ is quite lower than m_{ttl}.

(2) *Socially Selfish (SS) node:* a SS node forwards its local messages but accepts relaying non-local messages based on their social tie information. Thus, we define the overall utility of message m to a SS node sel as:

$$U_{sel}(m, \alpha_{sel}) = (1 - \alpha_{sel})U^{Ind}_{sel}(m) + \alpha_{sel}U^{Soc}_{sel}(m) \quad (2)$$

where $U^{Ind}_{sel}(m) \in [0, 1]$ and $U^{Soc}_{sel}(m) \in [0, 1]$ are respectively the individual and social utility of relaying m to sel, and $\alpha_{sel} \in (0, 0.5]$ is the *social-awareness degree* of sel. In Eq. (2), $\alpha_{sel} = 0$ implies that sel is an IS node and $\alpha_{sel} = 0.5$ implies that sel cares about the utility of its social ties as its individual utility. As a sender, sel forwards messages in M_{sel} in a descending order of their overall utility. As a receiver, sel accepts storing and relaying message m if $U_{sel}(m, \alpha_{sel}) > \partial$. Otherwise, sel drops m but generates a fake CR to hide its message dropping. We calculate the social utility of m to sel as follows:

$$U^{Soc}_{sel}(m) = \frac{ST_{sel,m_{src}} + ST_{sel,l}}{2} \quad (3)$$

where $ST_{sel,m_{src}}$ is the social tie strength between nodes sel and m_{src} and $ST_{sel,l}$ is the social tie strength between nodes sel and the sender of m that is denoted as node l. Based on Eq. (3), $U^{Soc}_{sel}(m) = 1$ if m is a local message ($m_{src} = sel$); otherwise, $0 \le U^{Soc}_{sel}(m) \le 1$.

3.5. Malicious node model

M nodes diffuse wrong watchdog information to their encountered nodes to disrupt the selfish node detection mechanism, but they participate in message relaying like W nodes. However, M nodes must have observed the behavior of other nodes or obtain true indirect watchdog information from W nodes to be able to generate proper wrong watchdog information. Nevertheless, the nodes do not know the type of each other. Thus, an M node cannot distinguish if the sender of particular watchdog information is a W or M node. Consequently, we design the M nodes in a way they generate random watchdog information about other nodes. Thus, an M node $mal \in M$ disseminates random watchdogs $P^{mal}_{i,FC}$, $P^{mal}_{i,IS}$, and $P^{mal}_{i,SS}$ about the reputation of node i to its encounters. The widely used notations are summarized in Table 1.

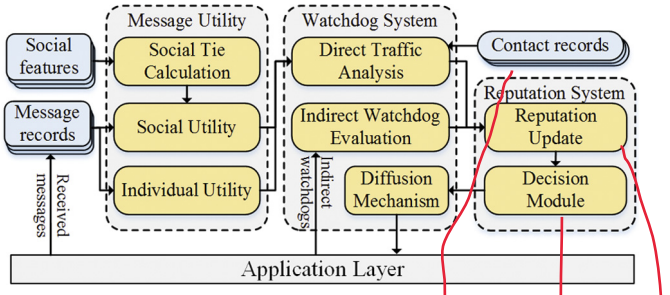


Fig. 1. The architecture of SoWatch.

4. Our proposed SoWatch scheme

In this section, we present an overview of SoWatch, followed by a detailed explanation of its components.

4.1. Overview

Fig. 1 depicts the architecture of SoWatch that includes three components: Message Utility, Watchdog System, and Reputation System. In the Message Utility, W node wch determines the individual and social utility of messages received from its' encountered node i (Eqs. (1) and (3)). In the Watchdog System, wch analyzes the messages based on their utility to identify i's possible selfishness type (*i.e.,* IS or SS). In case i is detected as selfish, wch explores the i's contact records to detect its message dropping. In addition, wch receives watchdog information (*i.e.,* the positives and negatives) about i from other nodes. Finally, wch updates the reputation of i based on the direct and indirect watchdog information. However, i might be an M node and provide false watchdogs about others to wch. To protect SoWatch against false watchdogs, we apply an evaluation mechanism in which wch validates the truthfulness of its received indirect watchdog information before using them. Besides, we design a diffusion mechanism, based on which wch can efficiently disseminate its positive and negatives to its encountered nodes.

4.2. Social tie calculation

In SoWatch, a W node explores the social tie information of their encountered nodes to deduce their routing behavior and identify IS and SS nodes. One major reason is that the nodes' social ties and relationships are relatively stable over a long time. For example, people in a family or a workplace have stable relationships and similar social features. Meanwhile, they willing cooperate with each other in message relaying. Based on this idea, we use social similarity to calculate the social tie strength between the nodes, based on which nodes with similar interests, backgrounds, and preferences contact each other more frequently (homophily [33]). Meanwhile, it is shown that the social similarity favors the cooperation of nodes in OMNs [34].

To calculate the social tie strength, we assume that encountered nodes i and j exchange their feature table (FT_i and FT_j) with each other. We apply a modified version of Jaccard coefficient [35] to measure the social similarity between two nodes. The main reason we chose the social similarity is that the simultaneous proximity of nodes' co-location and social information (*e.g.,* visited locations, common friends and similar interests) are considered in calculating the strength of social ties. We calculate the strength of social tie between nodes i and j as follows:

$$ST_{i,j} = \frac{\sum_{b=1}^{k} \delta_{i,j}(b) \times Sim_{i,l}(b)}{\sum_{b=1}^{k} \delta_{i,j}(b)} \quad (4)$$

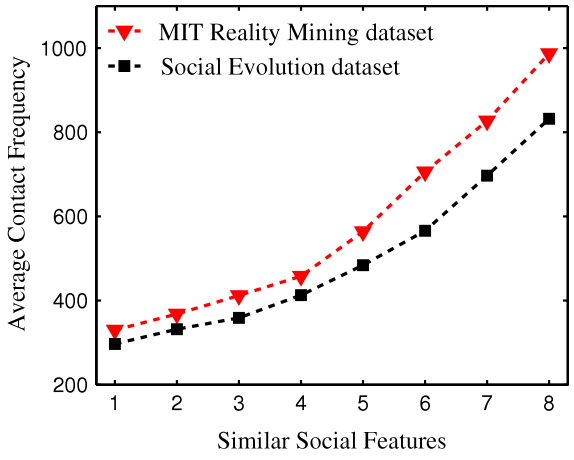


Fig. 2. The average number of Bluetooth contacts between the nodes according to their similar social features.

where $\delta_{i,j}(b) = 1$ if the value of feature b in both nodes i and j is not missing; Otherwise, $\delta_{i,j}(b) = 0$. In addition, $Sim_{i,j}(b) \in [0, 1]$ represents the similarity of nodes i and j regarding feature b, which is calculated as:

$$Sim_{i,j}(b) = \frac{|w_{i,b} \times f_{i,b} \cap w_{j,b} \times f_{j,b}|}{|f_{i,b} \cup f_{j,b}|} \tag{5}$$

where $f_{i,b} \in F_i$ and $f_{j,b} \in F_l$ are feature b of nodes i and j, respectively. In addition, $w_{i,b}$ is a weight factor that determines the importance of feature b of node i. We employ a unification process [36] to identify the importance of each feature. The main reason we apply the weight factor is that the significance of each social feature for different nodes can be different. For example, similar interests for a node might be more important than its direct contacts. We calculate the weight factor of feature b of node i as follows:

$$w_{i,b} = \frac{\frac{f_{i,b}}{\sum_{a=1}^{N} f_{a,b}}}{\frac{f_{i,1}}{\sum_{a=1}^{N} f_{a,1}} + \frac{f_{i,2}}{\sum_{a=1}^{N} f_{a,2}} + \cdots + \frac{f_{i,k}}{\sum_{a=1}^{N} f_{a,k}}} \tag{6}$$

where $\sum_{a=1}^{k} w_{i,a} = 1$, which implies that the summation of the weights of the features of node i equals 1.

We employ MIT Reality [37] and Social Evolution [38] datasets to validate the social tie calculation method. Both the datasets include the nodes' social and contact information (See Section 5.1 for more details). As shown in Fig. 2, the number of direct contacts between the nodes increases, as the number of their similar social features increases. For example, nodes with two similar features contact each other 368 and 332 times during the evaluation time on the MIT Reality and Social Evolution datasets, respectively, where nodes with seven similar social features contact each other 827 and 697 on the MIT Reality and Social Evolution datasets, respectively. Thus, our social tie calculation method can well represent the social interactions between the mobile carriers.

4.3. Watchdog system

In this section, we first describe the Direct Traffic Analysis module in which W node wch analyzes the messages received from its encountered node to identify its possible selfishness. We then explain the Indirect Watchdog Evaluation, based on which wch validates the indirect watchdog information received from another node. Finally, we explain the Diffusing Mechanism in which wch disseminates its watchdog information to its encountered nodes according to diffusion factors.

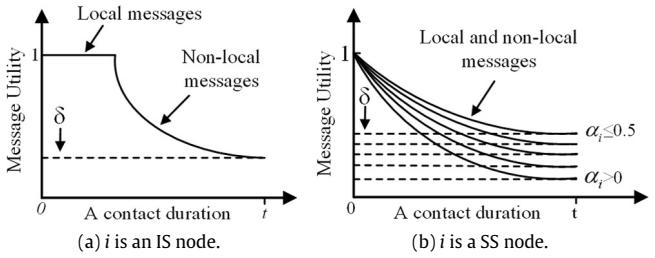


Fig. 3. The utility of messages forwarded by node i during a contact with respect to their forwarding order when (a) i is an IS node and (b) i is a SS node (α_i is the i's social-awareness degree and ∂ is a threshold value).

4.3.1. Direct traffic analysis

In this module, node wch analyzes the messages received from its encountered node i to identify its possible selfishness (*i.e.*, IS and SS). In case i is detected as a SS node, wch calculates its social-awareness degree α_i. Considering our selfishness model (Section 3.4), IS and SS nodes forward their messages in a descending order of their utility with the aim of maximizing their routing performance. Thus, analyzing the utility of messages forwarded by i based on their forwarding order can help wch determine the possible selfishness of i.

Fig. 3 demonstrates the pattern of the utility of messages forwarded by selfish node i based on their forwarding order. Fig. 3(a) shows the case when i is an IS node, and thus it first forwards its local messages (*i.e.*, $m_{src} = i$), which have the highest utility $U_i^{Ind}(m) = 1$ (Eq. (2)). Next, i forwards its non-local messages in a descending order of their individual utility $U_{sel}^{Ind}(m)$. Thus, analyzing the pattern of the utility of forwarding messages based on their forwarding order can help wch to identify if i is an IS node.

In contrast to IS nodes, detecting an SS node, say i, and its social-awareness degree with respect to the utility of its forwarding messages is a non-trivial problem. This is because SS nodes forward their messages based on their individual and social utility (Eq. (2)). Nevertheless, it is seen in Fig. 3(b) that the utility of messages forwarded by i decreases based on their forwarding order. In other words, node i forwards its messages in a descending order of $U_i(m, \alpha_i)$. However, an SS node may assign the highest utility to its non-local messages, whereas an IS node always assign the highest priority to its local messages. Consequently, wch can differentiate between IS and SS nodes based on the properties of their forwarding messages.

Based on the discussion above, W node wch should calculate the social-awareness degree of i (α_i) to be able to determine the utility of its forwarding messages (see Eq. (2)). We design Algorithm 1 to calculate the social-awareness degree of node i where $M_{wch,i}^t$ and FT_{wch} are the inputs of the algorithm and $0 < \alpha_i \leq 0.5$ is the output. Specifically, wch checks the utility of the messages in $M_{wch,i}^t$ to calculate α_i. If i is the source of all messages in $M_{wch,i}^t$, then $\alpha_i = 0$ that implies that i only forwards its local messages, and thus it does not provide information about its social-awareness degree (line 6). Otherwise, α_i is initiated by a small positive integer and then updated for each message $m \in M_{wch,i}^t$ as follows:

$$\alpha_i = \min\left(0.5, avg\left(\alpha_i, \frac{U_i^{Soc}(m)}{U_i^{Soc}(m) + U_i^{Ind}(m)}\right)\right) \tag{7}$$

where $U_i^{Soc}(m)$ and $U_i^{Ind}(m)$ are the social and individual utility of m to i, respectively. In Eq. (7), α_i increases as $U_i^{Soc}(m)$ increases that implies that the social-awareness of i increases when i forward messages received from nodes with whom it has strong social ties. The time complexity of Algorithm 1 is $O(|M_{wch,i}^t|)$, which is the number of messages in $M_{wch,i}^t$.

Algorithm 1 Social-awareness Degree Calculation

1: Node wch contacts node i at time t
2: **Input:** $M^t_{wch,i}$, FT_{wch};
3: **Output:** α_i; ▷ the social-awareness degree of i
4: **Initialize:** $\alpha_i = \textbf{null}$;
5: **for** each $m \in M^t_{wch,i}$ **do**
6: **if** $src_m == i$ **then**
7: $\alpha_i = 0$;
8: Continue;
9: **else**
10: Break;
11: **end if**
12: **end for**
13: **if** $\alpha_i == \textbf{null}$ **then**
14: $\alpha_i = \pounds$; ▷ $\pounds$ is the smallest positive number
15: **end if**
16: **for** each $m \in M^t_{wch,i}$ **do**
17: Update α_i based on m; ▷ Eq. 7
18: **end for**
19: **return** α_i

Algorithm 2 Direct Traffic Analysis

1: wch receives a set of messages from i at time t
2: **Input:** $M^t_{wch,i}$, FT_{wch};
3: **Output:** $P^{wch,t}_{i,FC}$, $P^{wch,t}_{i,IS}$, $P^{wch,t}_{i,SS}$;
4: **Initialize:** $P^t_{i,FC} = P^t_{i,IS} = P^t_{i,SS} = 0$;
5: **Initialize:** $msgTemp$ = **null**, $isUtilDesc$ = **true**;
6: Calculate α_i; ▷ Algorithm 1
7: **for** each $m \in M^t_{wch}$ **do**
8: **if** $msgTemp$ == **null then**
9: $msgTemp = m$;
10: Continue;
11: **end if**
12: **if** $U_i(m, \alpha_i) > U_i(msgTemp, \alpha_i)$ **then**
13: $isUtilDesc$ = **false**;
14: Break;
15: **else**
16: $msgTemp = m$;
17: **end if**
18: **end for**
19: **for** each $m \in M^t_{wch}$ **do**
20: **if** !$isUtilDesc$ **then**
21: Update $P^{wch,t}_{i,FC}$ using Eq. 12;
22: **else if** $isUtilDesc$ and $src_m == i$ **then**
23: Update $P^{wch,t}_{i,IS}$ using Eq. 13;
24: **else if** $isUtilDesc$ and $src_m \neq i$ **then**
25: Update $P^{wch,t}_{i,SS}$ using Eq. 14;
26: **end if**
27: **end for**
28: **return** $P^t_{i,FC}$, $P^t_{i,IS}$, $P^t_{i,SS}$

Once α_i is identified, wch can deduce i's routing preferences based on the utility of its forwarding messages. We design a message traffic analysis method, as shown in Algorithm 2 , in which wch identifies i's reputation ($P^{wch,t}_{i,FC}$, $P^{wch,t}_{i,IS}$, and $P^{wch,t}_{i,SS}$) based on the utility of messages in $M^t_{wch,i}$ at time t. In this algorithm, wch first calculates α_i using Algorithm 1 . Next, it checks the utility of messages in M^t_{wch} and sets $isUtilDesc$ =false if the utility of messages in M^t_{wch} based on their forwarding order is not descending (lines 6–16). In this case, $isUtilDesc$ = true implies that i exhibits selfish behavior in data forwarding. Finally, $P^{wch,t}_{i,FC}$, $P^{wch,t}_{i,IS}$, and $P^{wch,t}_{i,SS}$ are updated as follows:

$$P^{wch,t}_{i,FC} = P^{wch,t}_{i,FC} + \frac{(1 - U_i(m, \alpha_i))}{|M^t_{wch}|} \tag{8}$$

$$P^{wch,t}_{i,IS} = P^{wch,t}_{i,IS} + \frac{U^{Ind}_i(m)}{|M^t_{wch}|} \tag{9}$$

$$P^{wch,t}_{i,SS} = P^{wch,t}_{i,SS} + \frac{U^{Soc}_i(m)}{|M^t_{wch}|}. \tag{10}$$

The time complexity of Algorithm 2 is $O(|M^t_{wch,i}|)$, which equals the number of messages in $M^t_{wch,i}$.

4.3.2. Indirect watchdog evaluation

In addition to the direct traffic analysis, W node wch may receive indirect watchdog information about node i from other nodes. However, wch may contact an M node and receive wrong watchdog information about i that can disrupt the detection process. Thus, we design the Indirect Watchdog Evaluation module, based on which wch evaluates the trustfulness of its received watchdogs before applying them.

We assume that $P^{wch}_{i,FC}$, $P^{wch}_{i,IS}$, and $P^{wch}_{i,SS}$ are the current reputations of node i observed by wch. In addition, $P^{j,t}_{i,FC}$, $P^{j,t}_{i,IS}$, and $P^{j,t}_{i,SS}$ are the reputations of i provided by j at time t. Thus, wch accepts updating its opinion about i based on j's watchdogs if:

$$|P^{wch,t}_{i,FC} - P^{j,t}_{i,FC}| + |P^{wch}_{i,IS} - P^{j,t}_{i,FC}| + |P^{wch}_{i,SS} - P^{j,t}_{i,SS}| < 1. \tag{11}$$

4.3.3. Watchdog diffusion mechanism

Exchanging watchdog information between W nodes can improve the performance of a watchdog system in terms of detection time and accuracy. Nevertheless, diffusing all the positives and negatives can increase the network overhead significantly. Assuming that the number of W nodes is higher than S nodes, diffusing positive detections (*i.e.*, the reputation of detected IS and SS nodes) can result in a low communication overhead. Meanwhile, diffusing a part of negatives (*i.e.*, the reputation of detected FC nodes) can neutralize the effect of the false positives. Based on this idea, we define two parameters *positive diffusion factor* $df^{pos} \in [0, 1]$ and *negative diffusion factor* $df^{neg} \in [0, 1]$ to control the diffusion ratio of the positives (IS and SS detections) and negatives (FC detections). For example, if $df^{pos} = 0.5$ and $df^{neg} = 0$, W nodes only transmit half of their positives to their encountered nodes.

4.4. Reputation system

In the Reputation System, W nodes update their opinions about other nodes and identify their type.

4.4.1. Reputation update

In this module, W node wch updates the reputation of node i based on the direct and indirect watchdog information (received from the Direct Traffic Analysis and Indirect Watchdog Evaluation modules, respectively).

We let $P^{wch,t-\sigma}_{i,FC}$, $P^{wch,t-\sigma}_{i,IS}$, and $P^{wch,t-\sigma}_{i,SS}$ denote the reputation of i observed by wch before time t. Then, wch updates its opinion about i based on its observed direct watchdogs $P^{wch,t}_{i,FC}$, $P^{wch,t}_{i,IS}$, and $P^{wch,t}_{i,SS}$ at time t as:

$$P^{wch}_{i,FC} = \frac{P^{wch,t-\sigma}_{i,FC} + P^{wch,t}_{i,FC}}{2} \tag{12}$$

$$P^{wch}_{i,IS} = \frac{P^{wch,t-\sigma}_{i,IS} + P^{wch,t}_{i,IS}}{2} \tag{13}$$

$$P^{wch}_{i,SS} = \frac{P^{wch,t-\sigma}_{i,SS} + P^{wch,t}_{i,SS}}{2}. \tag{14}$$

Table 2
The type of node i based on its reputation.

Node Type	Reputations
Cooperative (FC)	$P_{i,FC} > P_{i,IS} + P_{i,SS}$
Individually Selfish (IS)	$P_{i,IS} > P_{i,SS}$ and $P_{i,IS} > P_{i,FC}$
Socially Selfish (SS)	$P_{i,SS} > P_{i,IS}$ and $P_{i,SS} > P_{i,FC}$
NoInfo	otherwise

While the direct watchdog information are more trustable, the indirect watchdogs provided by other nodes might be inaccurate or manipulated. Hence, we assign a weight factor to control the effects of the direct indirect watchdogs. Thus, wch updates the reputation of i based on $P^{j,t}_{i,FC}$, $P^{j,t}_{i,IS}$, and $P^{j,t}_{i,SS}$ diffused by another node, say node j, at time t as:

$$P^{wch}_{i,FC} = \delta P^{wch,t-\sigma}_{i,FC} + (1-\delta)P^{j,t}_{i,FC} \tag{15}$$

$$P^{wch}_{i,IS} = \delta P^{wch,t-\sigma}_{i,IS} + (1-\delta)P^{j,t}_{i,IS} \tag{16}$$

$$P^{wch}_{i,SS} = \delta P^{wch,t-\sigma}_{i,SS} + (1-\delta)P^{j,t}_{i,SS} \tag{17}$$

where δ is a weight factor that identifies the importance of the direct and indirect watchdog information received by node wch on the calculation of the reputation of node i. In the experiments, we set $\delta = 0.9$ to achieve the highest detection performance.

4.4.2. Decision module

Using the Decision Module, node wch identifies the type of other nodes based on their reputation. In particular, wch identifies if sample node i is either cooperative or selfish. Then, wch determines i's selfishness type in case it is detected as selfish. We compare i's reputations to identify its type. As shown in Table 2, if the FC reputation of i is higher than the summation of its IS and SS reputations, then i is FC; and if the IS reputation of i is higher than its SS and FC reputations, then i is IS; and if the SS reputation of i is higher than its IS and FC reputations, then i is SS. Finally, if the reputations of i do not provide sufficient information about its routing behavior, its status is *NoInfo*.

Once a selfish node is detected, wch investigates its contact records to identify its dropping messages. As discussed in Section 3.1, when a node contacts another node, it generates a CR with a sequential number where a CR with the higher sequential number has a bigger timestamp. Thus, dropping messages or manipulating CRs by the selfish nodes violate the consistency of their CRs, based on which wch can identify the messages dropped by its encountered nodes.

5. Performance evaluation

We evaluate the performance of SoWatch using Opportunistic Network Environment (ONE) simulator [39], which is a trace-driven simulator to evaluate DTN protocols. We have two main goals in the experiments. First, we assess the performance of SoWatch in terms of some important performance metrics. Second, we make a comparison between SoWatch and a benchmark contact-based watchdog [19] in which each node collects the contact records about other nodes from its encountered nodes. Next, each node investigates the consistency of its collected contact records to detect message droppers. The main reason we compare SoWatch with the contact-based detection method in [19] is to explore the role of nodes' social information in their cooperative behavior on message relying and resource sharing. In addition, we aim at investigating the advantages of exploiting nodes' social features on detecting their selfish behavior in message relaying in terms of different performance metrics.

Table 3
The main features of the datasets.

Dataset	Reality mining	Social evolution
Device	Nokia 6600	Smartphone
Year	2004	2009
No. nodes	106	80
Duration (day)	246	240
Granularity (sec)	300	360
No. Blue. contacts	1 259 148	2 124 565
Avg. Blue. contacts	5118.45	8852.35

5.1. Real-world datasets

We use two datasets, MIT Reality [37] and Social Evolution [38], provided by the MIT Human Dynamics lab (hd.media.mit.edu) to set the nodes' contacts and social features. Each dataset includes the hashed MAC address and sensor data of devices as well as the survey data about the participants. The sensor data includes information such as the Bluetooth proximity, cellular tower IDs, and phone call logs. Furthermore, the survey data include the participants' friendship information, interests, and questionnaires. We process each dataset in MATLAB and convert them to a compatible format to import to ONE. The properties of each dataset can be described as follows:

MIT Reality: is collected over 8 months on the MIT campus in which 106 users carry Nokia 6600 smartphones. In the simulations, we filter 88 users with sufficient Bluetooth contacts and social features between October 2004 and April 2005. We select 5 features in this dataset to identify the social tie strength between the nodes as follows: (1) contact frequency, (2) job or affiliation, (3) visited locations (WLAN data), (4) phone calls, and (5) SMS logs.

Social Evolution: this dataset consists of the traces of 80 undergraduate students who carry their cell phones for 8 months. In the experiments, we select the contacts and social features of 74 users between January and June 2009. We use 6 features to measure the social ties as follows: (1) contact frequency, (2) user interests (*e.g.*, music, politic), (3) living sector, (4) year in school, and (5) phone calls, and (6) SMS logs. Table 3 summarizes the main features of the datasets.

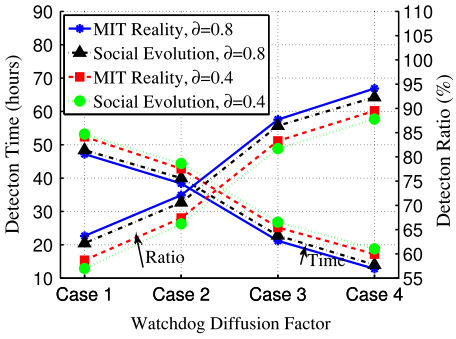
5.2. Simulation settings and performance metrics

We setup a wireless network in which nodes contact each other via Bluetooth interface with a bandwidth of 5 Mbps. Each node generates messages with size 0.5∼1 Megabyte (MB) in uniform interval 5∼10 h. In addition, the buffer capacity of each node is set to 30 MB and the TTL of messages is 5 days. We present the averaged results for each experiment where each experiment is run 10 times with different seeds to provide highly confident results. We assume that the nodes only store and exchange their most recent 1000 CRs.

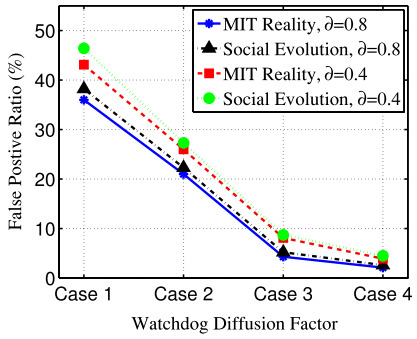
We apply Spray-and-Wait (SnW) [40] algorithm as the underlying routing protocol in SoWatch, which is already implemented in the ONE simulator. In SnW, the source can generate maximum L copies of each message, and when it contacts intermediate node i, it gives half of the message copies (*i.e.*, $\frac{L}{2}$) to i (if $L > 1$). The same mechanism is applied when i contacts another encountered node until each node (the source and the relays) has more than one copy. Finally, L nodes will hold a copy of the message and wait until one of them can deliver the message to its destination.

We evaluate four metrics in the simulations:

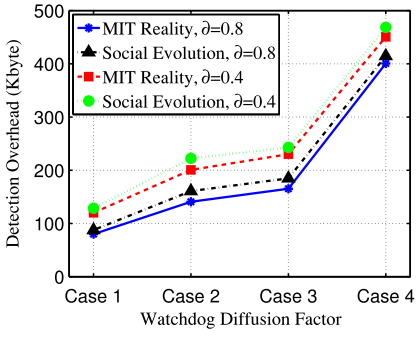
- **Detection time**: the average time that it takes a W node detects an S node.
- **Detection ratio**: the average percentage of S nodes detected by W nodes over a time period.



(a) Detection time and ratio.



(b) False positive detection ratio.



(c) Detection overhead.

Fig. 4. The performance of SoWatch with different watchdog diffusion factors.

- **False positive detection ratio**: the average percentage of W nodes that are wrongly detected as S nodes by W nodes over a time period.
- **Detection overhead**: the average number of bytes of control packets (*e.g.*, contact records, social features, and watchdog information) exchanged between encountered nodes.

5.3. Evaluation results

5.3.1. Varying the watchdog diffusion factors

In Fig. 4, we evaluate the impact of the positive and negative diffusion factors (df^{pos} and df^{neg}) on SoWatch over the Reality Mining and Social Evolution datasets. As shown in Table 4, we consider four setting for the diffusion factors in these experiments. We set 80% of W nodes, 10% of S nodes (including 5% of IS nodes and 5% of SS nodes with $\alpha_i = 0.3$), and 10% of M nodes.

Fig. 4(a) shows the impact of the diffusion factors on the detection time and ratio. In general, as the diffusion factors increase, the detection ratio represents a continuous upward trend on both the datasets. This occurs because disseminating a large volume of the watchdog information helps W nodes to detect S nodes accurately. We also observe that when the diffusion factors vary from Case 2 to Case 3, the detection ratio grows sharply. For example, when it changes from Case 2 to Case 3, the detection ratio over the Reality Mining increases from 72.1% to 87.7% for $\partial = 0.8$ and from 67.4% to 83.3% for $\partial = 0.4$. The reason is that diffusing a high ratio of positive detections help W nodes identify a high number of S nodes. Meanwhile, diffusing a small ratio of negatives neutralizes the effect of false positives. In contrast to the detection ratio, the detection time decreases over the entire period continuously, as the diffusion factors increase. For example, when the diffusion factor changes from Case 1 to Case 2, the detection time on the Reality Mining decreases from 47.2 to 38.3 h for $\partial = 0.8$ and from 52.4 to 42.8 h for $\partial = 0.4$. Furthermore, a sharp downward trend is seen in the detection time when the diffusion factors vary from Case 2 to Case 3. The reason is that diffusing a large volume of positives shortens the detection time.

Fig. 4(b) illustrates the false positive detection ratio when the diffusion factors vary between cases 1, 2, 3, and 4. The trend shows that the ratio of the false positive detections drops constantly as the diffusion factors increase. For example, when the diffusion factors shift from Case 2 to Case 3, the false positive ratio using the Social Evolution decreases from 22.3% to 5.2% for $\partial = 0.8$ and from 27.3% to 8.7% for $\partial = 0.4$. The reason is that diffusing a high ratio of positive detections helps W nodes to update their opinion about others more accurately. It is also observed that the false positive detection ratio declines gradually when the diffusion factors change from Case 3 to Case 4, while this metric shows a rapid decrease when the diffusion factors shifts from Case 1 to Case 2 and from Case 2 to Case 3. Thus, it can be concluded that diffusing a high volume of negative detections in Case 4 does not decrease the false positive detections considerably in comparison with Case 3.

Fig. 4(c) shows the impact of different diffusion ratios on the detection overhead. It can be seen that the detection overhead increases continuously when the diffusion factors vary between cases 1, 2, 3, and 4. The reason is that the number of watchdog packets W nodes disseminate through the network increases. It is also observed that the trend from Case 1 to Case 2 as well as Case 3 to Case 4 depicts a rapid increase, while the trend from Case 2 to Case 3 shows a moderate growth. For example, when the diffusion factors change from Case 2 to Case 3, the communication overhead on the Reality Mining increases from 140.7 to 165.2 kilobytes, while the cost rises dramatically from 165.2 to 400.6 kilobytes when the diffusion factors change from Case 3 to Case 4. The reason is that W nodes disseminate large volumes of their positive and negative detections in Case 4. In summary, it can be observed that diffusing a large volume of the positive detections along with a small portion of the negative detections results in a better tradeoff in SoWatch. Consequently, in the rest of the experiments, we set the diffusion factors in SoWatch to Case 3 in order to achieve the highest tradeoffs.

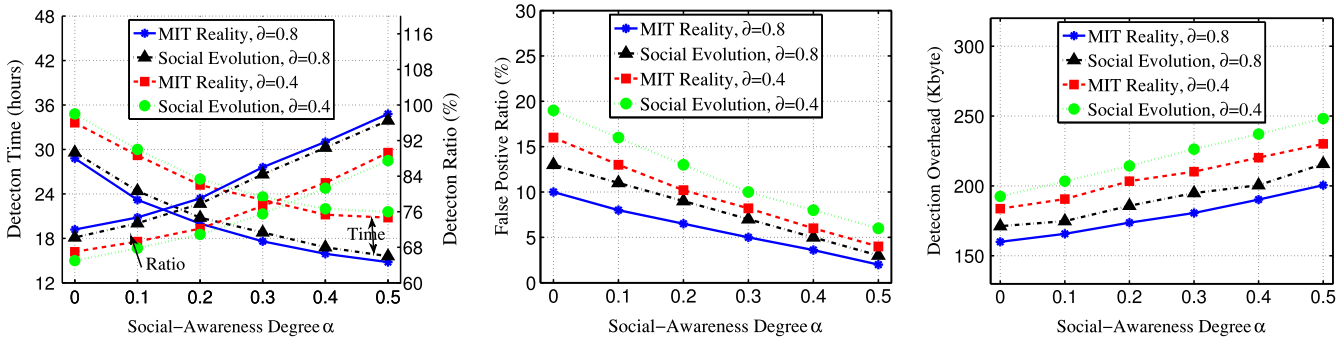
5.3.2. Varying the social-awareness degree

Fig. 5 shows the impact of the social-awareness degree of the SS nodes on the performance of SoWatch when it varies between 0 and 0.5. We use 80% of W nodes, 10% of SS nodes with $\partial = 0.8$, and 10% of M nodes. In addition, the diffusion factors are set to Case 3.

Fig. 5(a) displays the effects of different values of α_{sel} for each $sel \in S$ on the detection time and ratio. In common, the detection ratio rises continuously as α_{sel} increases. For example, when α_{sel} increases from 0.2 to 0.3, the detection ratio on the Reality Mining increases from 79.1% to 86% when $\partial = 0.8$ and from 72.2% to 77.3% when $\partial = 0.4$. This occurs because S nodes forward messages with high social utilities that help W nodes to detect them precisely.

Table 4
Different settings for the watchdog diffusion factors.

Setting	Description
Case 1	low positive diffusion factor ($df^{pos} = 25\%$) low negative diffusion factor ($df^{neg} = 25\%$)
Case 2	low positive diffusion factor ($df^{pos} = 25\%$) high negative diffusion factor ($df^{neg} = 75\%$)
Case 3	high positive diffusion factor ($df^{pos} = 75\%$) low negative diffusion factor ($df^{neg} = 25\%$)
Case 4	high positive diffusion factor ($df^{pos} = 75\%$) high negative diffusion factor ($df^{neg} = 75\%$)



(a) Detection time and ratio. (b) False positive detection ratio. (c) Detection overhead.

Fig. 5. The performance of SoWatch for different value of the social-awareness degree.

In contrast, the detection time decreases over the entire period continuously when α_{sel} increases. For example, when α_{sel} grows from 0.3 to 0.4, the detection time on the Reality Mining declines from 17.6 to 15.9 h when $\partial = 0.8$ and from 23.2 to 21.2 when $\partial = 0.4$. The reason behind is that forwarding messages with the high social utility by S nodes help W nodes to identify their type swiftly.

Fig. 5(b) illustrates the results in terms of the false positive detection ratio. It can be seen that the false positive detection ratio declines gradually when α_{sel} increases. For example, when α_{sel} increases from 0.2% to 0.3%, this metric on the Social Evolution falls from 9.1% to 7.2% when $\partial = 0.8$ and from 13.2% to 10.1% when $\partial = 0.4$. The reason behind is that when W nodes receive messages with the high social utility from other nodes, they can deduce the routing objective of their encountered nodes and thus, distinguish between S and non-selfish nodes precisely. Meanwhile, the figure demonstrates that the false positive detection ratio has the best performance over the Reality Mining when $\partial = 0.8$, whereas this metric on the Social Evolution has the worst results when $\partial = 0.4$.

Fig. 5(c) reveals the impact of α_{sel} on the detection overhead. The figure depicts a ceiling trend in all the settings as α_{sel} increases. Meanwhile, we observe that SoWatch achieves the lowest detection overhead over the Reality Mining when $\partial = 0.8$ while this metric on the Social Evolution has the highest overhead when $\partial = 0.4$. The reason is that as α_{sel} increases, S nodes forward messages with the high social utility to their encountered nodes. Thus, W nodes can detect a large number of S nodes and hence, disseminate a high volume of their watchdog detections to other nodes that increase the detection overhead.

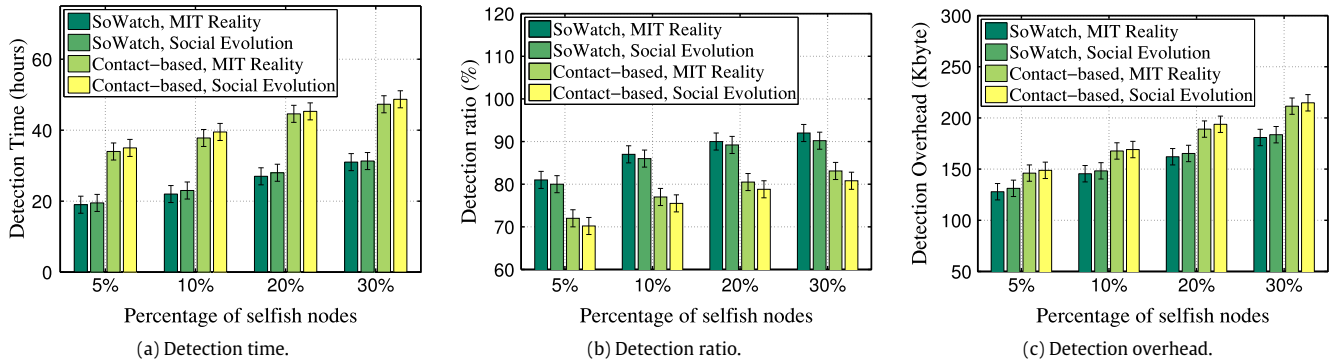
5.3.3. *Varying the number of selfish nodes*

In Fig. 6, we benchmark SoWatch against the contact-based watchdog when the percentage of selfish nodes varies between 5% and 30%. For each S node in SoWatch, we set $\alpha_i = 0.3$ and $\partial = 0.8$. Furthermore, we set 10% of M nodes and the rest of the nodes are W nodes. In addition, we set the diffusion factors to Case 3. The standard deviations are shown using lines. Note that in the rest of the experiments, we do not evaluate the false positives because the contact-based watchdog does not have false positive (see [19]).

Fig. 6(a) compares the algorithms in the detection time. We observe that the detection time in SoWatch is considerably shorter than the contact-based watchdog on both the datasets. For example, when 20% and 30% of the nodes are selfish, SoWatch outperforms the contact-based watchdog on the MIT Reality by 49% and 41%, respectively. This occurs because W nodes in SoWatch detect the selfish nodes based on their stable social tie information. While, the watchdog nodes in the contact-based watchdog need to collect multiple versions of the contact records about other nodes from different nodes to be able to check their consistency and detect the selfish nodes that result in long detection time.

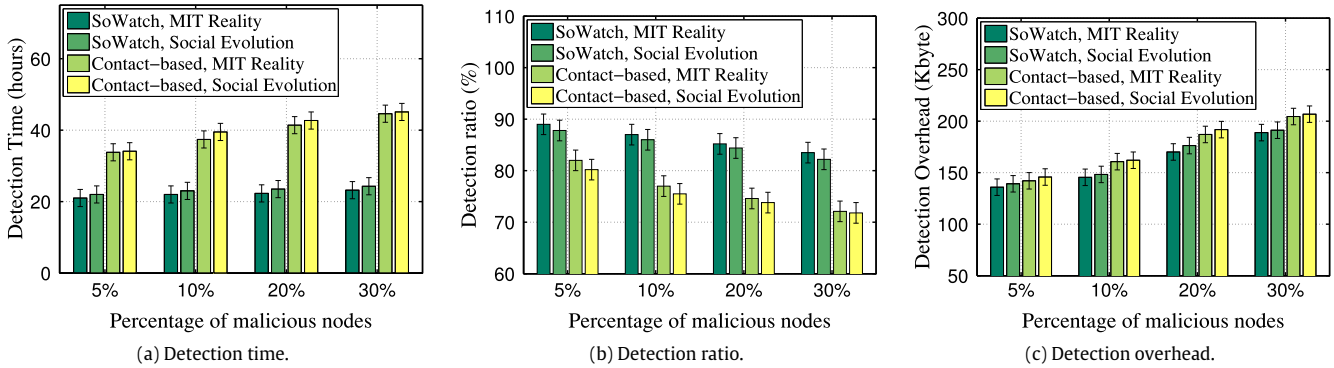
Fig. 6(b) compares the algorithms in terms of the detection ratio. In general, the detection ratio of the algorithms depicts a continuous downward trend as the percentage of selfish nodes increases. The reason is that watchdog nodes detect a large number of selfish nodes based on their direct or indirect watchdog information. Meanwhile, it is observed that SoWatch outperforms the contact-based watchdog in the detection ratio over both the datasets. For example, when 20% of nodes are selfish, the detection ratio in SoWatch over the Reality Mining and Social Evolution datasets is 9.5% and 10.4% higher than the contact-based watchdog, respectively. This is because W nodes in SoWatch share their observations about the routing behavior of other nodes with each other that help them to detect a large number of S nodes.

Fig. 6(c) demonstrates the evaluation results in the detection overhead. Overall, it is observed that the number of control packets disseminated to detect selfish nodes increases as the number of selfish nodes raises. Comparatively, we observe that SoWatch on the MIT Reality has the lowest detection overhead while the contact-based watchdog on the Social Evolution has the highest detection cost. For example, when 30% of the nodes are selfish,



(a) Detection time. (b) Detection ratio. (c) Detection overhead.

Fig. 6. The performance comparison of the algorithms with different number of selfish nodes.



(a) Detection time. (b) Detection ratio. (c) Detection overhead.

Fig. 7. The performance comparison of the algorithms with different number of malicious nodes.

the detection overhead of SoWatch over the Reality Mining and Social Evolution is 17.2% and 16.9% lower than the contact-based watchdog, respectively.

5.3.4. Varying the number of malicious nodes

In Fig. 7, we compare the algorithms when the percentage of malicious nodes varies between 5% and 30%. We set 10% of S nodes in SoWatch (including 5% of IS nodes and 5% of SS nodes with $\alpha_{sel} = 0.3$) where $\partial = 0.8$, and the rest of the nodes are W nodes. In addition, the diffusion factors are set to Case 3.

Fig. 7(a) indicates the performance of the algorithms in the detection time. We observe that as the number of the malicious nodes increases, the detection time in SoWatch is substantially shorter than the contact-based watchdog. For example, when the percent of the malicious nodes is 10% and 20%, the detection time in SoWatch outperforms the contact-based watchdog over the Reality Mining and Social Evolution by 52% and 59%, respectively. The reason is that W nodes in SoWatch well tolerate against the wrong watchdogs disseminated by evaluating the truthfulness of their received indirect watchdogs.

Fig. 7(b) compares the algorithms in the detection ratio. As a general trend, the detection ratio in both the algorithms decreases as the number of malicious nodes increases. Meanwhile, it is observed that the detection ratio in SoWatch shows a gradual decrease while this metric in the contact-based watchdog falls down rapidly. This is because W nodes in SoWatch can detect false positive and negative detections received from M nodes using their CRs accurately.

Fig. 7(c) demonstrates the evaluation results in the detection overhead. We observe that the detection overhead shows a continuous upward trend in both the algorithms as the number of malicious nodes increases. Furthermore, the detection overhead in SoWatch is lower than the contact-based watchdog through the whole trend. The reason behind is that watchdog nodes in the contact-based watchdog exchange their contact history with each other for protecting the detection scheme against the malicious nodes that increase its detection overhead.

6. Conclusion

In this paper, we proposed a social-based watchdog system (SoWatch) for OMNs in which the watchdog nodes analyze the messages received from their encountered nodes based on their social tie information. Meanwhile, the watchdog nodes cooperatively share their opinions about other nodes with each other to further improve the detection performance. Next, we designed a reputation system, based on which the watchdog nodes update the reputation of other nodes based on the direct and indirect watchdog information to identify selfish nodes and their selfishness type. Our experiments using real-world datasets demonstrated that SoWatch outperforms a benchmark contact-based watchdog system in terms of the detection time, detection ratio, and communication cost. In the future, we plan to design an incentive scheme in energy-constrained selfish OMNs.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group No. (RGP-1438-27). This work was partially supported by the National Natural Science Foundation of

China (61572106, 61772551), and the US National Science Foundation grants CNS-1355505, CCF-1533918, and CCF-1725755. We are grateful to the anonymous reviewers for their constructive suggestions to help us improve the quality of the manuscript.

References

[1] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proc. ACM SIGCOMM, 2003, pp. 27–34.

[2] Z. Lu, X. Sun, T.L. Porta, Cooperative data offloading in opportunistic mobile networks, in: Proc. IEEE INFOCOM, 2016, pp. 1–9.

[3] Z. Ning, F. Xia, N. Ullah, X. Kong, X. Hu, Vehicular social networks: Enabling smart mobility, IEEE Commun. Mag. 55 (5) (2017) 16–55.

[4] M.Y.S. Uddin, H. Ahmadi, T. Abdelzaher, R. Kravets, Intercontact routing for energy constrained disaster response networks, IEEE Trans. Mob. Comput. 12 (10) (2013) 1986–1998.

[5] N. Chakchouk, A survey on opportunistic routing in wireless communication networks, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2214–2241.

[6] F. Xia, L. Liu, B. Jedari, S. Das, PIS: A multi-dimensional routing protocol for socially-aware networking, IEEE Trans. Mob. Comput. 15 (11) (2016) 2825–2836.

[7] Q. Li, W. Gao, S. Zhu, G. Cao, A routing protocol for socially selfish delay tolerant networks, Ad Hoc Netw. (ISSN: 1570-8705) 10 (8) (2012) 1619–1632.

[8] F. Xia, B. Jedari, L.T. Yang, J. Ma, R. Huang, A signaling game for uncertain data delivery in selfish mobile social networks, IEEE Trans. Comput. Soc. Syst. 3 (2) (2016) 100–112.

[9] F. Xia, L. Liu, J. Li, J. Ma, A.V. Vasilakos, Socially-aware networking: A survey, IEEE Syst. J. 9 (3) (2015) 904–921.

[10] A. Keranen, M. Pitkanen, M. Vuori, J. Ott, Effect of non-cooperative nodes in mobile DTNs, in: Proc. IEEE WoWMoM 2011, pp. 1–7.

[11] Y. Li, G. Su, D.O. Wu, D. Jin, L. Su, L. Zeng, The impact of node selfishness on multicasting in delay tolerant networks, IEEE Trans. Veh. Technol. 60 (5) (2011) 2224–2238.

[12] S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proc. ACM MobiCom, 2000, pp. 255–265.

[13] Y. Li, P. Hui, D. Jin, L. Su, L. Zeng, Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks, IEEE Commun. Lett. 14 (11) (2010) 1026–1028.

[14] P. Serrpezis, T. Spyropoulos, Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks, Comput. Commun. 48 (2014) 71–83.

[15] E.M. Daly, M. Haahr, Social network analysis for routing in disconnected delay-tolerant MANETs, in: Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '07, 2007, pp. 32–40.

[16] Y. Guo, S. Schildt, L. Wolf, Detecting blackhole and greyhole attacks in vehicular Delay Tolerant Networks, in: Proc. COMSNETS, 2013, pp. 1–7.

[17] F. Li, J. Wu, A. Srinivasan, Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets, in: Proc. IEEE INFOCOM, 2009, pp. 2428–2436.

[18] T.N.D. Pham, C.K. Yeo, Detecting colluding blackhole and greyhole attacks in delay tolerant networks, IEEE Trans. Mob. Comput. 15 (5) (2016) 1116–1129.

[19] Q. Li, G. Cao, Mitigating routing misbehavior in disruption tolerant networks, IEEE Trans. Inf. Forensics Secur. 7 (2) (2012) 664–675.

[20] H. Zhu, S. Du, Z. Gao, M. Dong, Z. Cao, A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2014) 22–32.

[21] E. Hernández-Orallo, M.D.S. Olmos, J.C. Cano, C.T. Calafate, P. Manzoni, CoCoWa: A collaborative contact-based watchdog for detecting selfish nodes, IEEE Trans. Mob. Comput. 14 (6) (2015) 1162–1175.

[22] J.A.F.F. Dias, J.J.P.C. Rodrigues, F. Xia, C.X. Mavromoustakis, A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks, IEEE Trans. Ind. Electron. 62 (12) (2015) 7929–7937.

[23] Y. Cai, Y. Fan, D. Wen, An incentive-compatible routing protocol for two-hop delay-tolerant networks, IEEE Trans. Veh. Technol. 65 (1) (2016) 266–277.

[24] A. Kate, G.M. Zaverucha, U. Hengartner, Anonymity and security in delay tolerant networks, in: Proc. IEEE SecureComm, 2007, pp. 504–513.

[25] R.-I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, V. Cristea, SENSE: A collaborative selfish node detection and incentive mechanism for opportunistic networks, J. Netw. Comput. Appl. 41 (2014) 240–249.

[26] E. Jaho, M. Karaliopoulos, I. Stavrakakis, Social similarity favors cooperation: the distributed content replication case, IEEE Trans. Parallel Distrib. Syst. 24 (3) (2013) 601–613.

[27] B. Jedari, L. Liu, T. Qiu, A. Rahim, F. Xia, A game-theoretic incentive scheme for social-aware routing in selfish mobile social networks, Future Gener. Comput. Syst. 70 (2017) 178–190.

[28] K. Liu, J. Deng, P.K. Varshney, K. Balakrishnan, An acknowledgment-based approach for the detection of routing misbehavior in MANETs, IEEE Trans. Mob. Comput. 6 (5) (2007) 536–550.

[29] E. Hernández-Orallo, M.D. Serra. Olmos, J.-C. Cano, C.T. Calafate, P. Manzoni, Evaluation of collaborative selfish node detection in MANETs and DTNs, in: Proc. ACM MSWiM, 2012, pp. 159–166.

[30] X. Chen, X. Gong, L. Yang, J. Zhang, Exploiting social tie structure for cooperative wireless networking: A social group utility maximization framework, IEEE/ACM Trans. Netw. 24 (6) (2016) 3593–3606.

[31] Y. Sun, L. Yin, W. Liu, Defending sybil attacks in mobile social networks, in: Proc. IEEE INFOCOM, 2014, pp. 163–164.

[32] C. Bermejo, R. Zheng, P. Hui, An empirical study of human altruistic behaviors in opportunistic networks, in: Proc. 7th International Workshop on Hot Topics in Planet-scale mObile Computing and Online Social neTworking, 2015, pp. 43–48. ISBN 978-1-4503-3517-1.

[33] L.L.M. McPherson, J. Cook, Birds of a feather: Homophily in social networks, Ann. Rev. Sociol. 27 (1) (2001) 415–444.

[34] E. Jaho, M. Karaliopoulos, I. Stavrakakis, Social similarity favors cooperation: The distributed content replication case, IEEE Trans. Parallel Distrib. Syst. 24 (3) (2013b) 601–613.

[35] P. Jaccard, Etude comparative de la distribution florale dans une portion des Alpes et du Jura, in: Bulletin de la Société vaudoise des sciences naturelles, Impr. Corbaz, 1901.

[36] L. Gao, M. Li, A. Bonti, W. Zhou, S. Yu, Multidimensional routing protocol in human-associated delay-tolerant networks, IEEE Trans. Mob. Comput. 12 (11) (2013) 2132–2144.

[37] N. Eagle, A. Pentland, Reality mining: Sensing complex social systems, Pers. Ubiquitous Comput. 10 (4) (2006) 255–268.

[38] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, A. Pentland, Sensing the health state of a community, IEEE Pervasive Comput. 11 (4) (2012) 36–45.

[39] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: Proc. International Conference on Simulation Tools and Techniques, Simutools '09, 2009, pp. 1–10.

[40] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proc. ACM SIGCOMM, 2005, pp. 252–259.

Behrouz Jedari received the B.Sc. and M.Sc. degrees from the Islamic Azad University, Qazvin, Iran, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree with the School of Software, Dalian University of Technology, Dalian, China. His current research interests include heterogeneous wireless networks, resource management and allocation, and mobile social networks.

Feng Xia received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor in School of Software, Dalian University of Technology, China. He is the (Guest) Editor of several international journals. He serves as General Chair, PC Chair, Workshop Chair, or Publicity Chair of a number of conferences. Dr. Xia has published 2 books and over 200 scientific papers in international journals and conferences. His research interests include computational social science, network science, data science, and mobile social networks. He is a Senior Member of IEEE (Computer Society, SMC Society) and ACM (SIGWEB), and a Member of AAAS.

Honglong Chen received the M.E. degree in control theory and control engineering from Zhejiang University, China, in 2008, and the Ph.D degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2012. He was a Postdoctoral Researcher in the School of CIDSE at Arizona State University from 2015 to 2016. He is currently an Associate Professor with the College of Information and Control Engineering, China University of Petroleum, China. His current research interests are in the areas of RFID and Internet of Things. He has published more than 30 research papers in many international journals and conferences including IEEE TVT, IEEE TETC, IEEE IoTJ, IEEE INFOCOM, IEEE ICPP, IEEE ICCCN, etc. He is a member of IEEE and ACM.

Sajal K. Das is a professor of Computer Science and Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology, Rolla, USA. During 2008–2011, he served the US National Science Foundation as a Program Director in the Division of Computer Networks and Systems. His current research interests include theory and practice of wireless and sensor networks, mobile and pervasive computing, cyber–physical systems and smart environments including smart grid and smart healthcare, distributed and cloud computing, security and privacy, biological and social networks, applied graph theory and game theory. He has published more than 600 research articles in high quality journals and refereed conference proceedings, 52 invited book chapters, and coauthored 4 books. He holds 5 US patents and received 10 Best Paper Awards in prestigious conferences such as ACM MobiCom'99, IEEE PerCom'06, IEEE SmrtGridComm'12, and IEEE SmartComp14. He is also a recipient of numerous awards including the IEEE Computer Society's Technical Achievement Award for pioneering contributions to sensor networks and mobile computing, Lockheed Martin Teaching Excellence Award, and Graduate Dean's Award of Excellence. He is the founding Editor-in-Chief of the Pervasive and Mobile Computing journal, and an Associate Editor of IEEE Transactions on Mobile Computing, ACM Transactions on Sensor Networks, Journal of Parallel and Distributed Computing, and Journal of Peer to Peer Networking and Applications. He co-founded IEEE WoWMoM, IEEE PerCom, and IEEE SmartComp conferences, and served on numerous conference committees as General Chair, Program Chair, or Program Committee member. He is an IEEE Fellow.

Amr Tolba received the M.Sc. and Ph.D. degrees from the Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently an Associate Professor with the Faculty of Science, Menoufia University, Egypt. He is currently on leave from Menoufia University to Computer Science Department, Community College, King Saud University, Saudi Arabia. He serves as a Technical Program Committee Member in several conferences. He has authored/coauthored over 30 scientific papers in international journals and conference proceedings. His main research interests include socially-aware network, Internet of Things, intelligent systems, big data, recommender systems, and cloud computing.

Zafer ALMakhadmeh received the M.Sc. and Ph.D. degrees from Department of Computer Engineering, Faculty of Information and Computer Engineering, Kharkov National Technical University of Ukraine, in 1998 and 2001 respectively. He is currently an assistant professor at Computer Science Department, Community College, King Saud University, Saudi Arabia. His main research interests include cloud computing, social network analysis, big data, and intelligent systems.